



PTO/SB/21 (09-04)

Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**TRANSMITTAL
FORM**

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	10/082,794
Filing Date	February 22, 2002
First Named Inventor	Bau
Art Unit	2192
Examiner Name	Rutten, James D.
Attorney Docket Number	109870-130096

ENCLOSURES (Check all that apply)

<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Reply to Missing Parts/ Incomplete Application <input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____ <input type="checkbox"/> Landscape Table on CD	<input type="checkbox"/> After Allowance Communication to TC <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input checked="" type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): Return Receipt Postcard
Remarks		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Schwabe, Williamson & Wyatt, P.C.		
Signature			
Printed name	Robert C. Peck		
Date	March 9, 2006	Reg. No.	56,826

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Yvette L. Chriscaden	Date	March 9, 2006

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Docket No.: 109870-130096

MAIL STOP: APPEAL BRIEF-PATENTS

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

By: Yvette L. Chriscaden Date: March 9, 2006
Yvette L. Chriscaden

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Before the Board of Patent Appeals and Interferences

App. No. : 10/082,794 Confirmation No.: 2046
Inventor : David Bau III et al.
Filed : February 22, 2002
Title : ANNOTATION BASED DEVELOPMENT PLATFORM FOR
STATEFUL WEB SERVICES
Art Unit : 2192
Examiner : Rutten, James D.
Customer No. : 25,943

MAIL STOP: APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

**RESUBMISSION OF APPELLANT'S BRIEF IN SUPPORT OF APPELLANT'S
APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Dear Sir:

This is a re-submission of Appellant's Brief in response to the Non-Compliant Notice mailed on February 21, 2006. The deficiency has been corrected. This appeal furthers the Notice of Appeal filed on November 8, 2005. The appeal arises from a final decision by the Examiner in the final Office Action, dated August 10, 2005. The final decision was in response to arguments filed on May 9, 2005, in response to an earlier office action, mailed February 9, 2005.

Appellants submit this *Brief on Appeal* in triplicate. Payment in the amount of \$500.00 to cover the fee for filing the *Brief on Appeal* was tendered with the original submission. Appellants respectfully request consideration of this appeal by the Board of Patent Appeals and Interferences for allowance of the present patent application.

Real Party in Interest:

This application is assigned to BEA Systems, Inc., having a principal place of business at 2315 North First Street, San Jose, California 95131. The assignment is recorded at the United States Patent and Trademark Office, reel 012948, frame 0811.

Related Appeals and Interferences:

To the best of Appellants' knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

Status of Claims:

Appellants appeal the rejection of claims 1-52. Claims 1-52 were pending and were rejected in the Final Office Action dated August 10, 2005. Claims 1-52 are reproduced, as pending, in Appendix A.

Status of Amendments:

Appellants have made no amendments subsequent to the Examiner's final rejection.

Summary of the Claimed Subject Matter:

Independent claim 1 is directed towards *a method specifying a stateful web service within a procedural programming environment* that comprises "providing a source code representation of at least a portion of web service logic, the logic including one or more methods; identifying one of said one or more methods to be exposed as part of the stateful web service; and specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational

state related to the identified method.” Element 102 of Figure 1 illustrates one example of a web server performing the operations recited in claim 1. Element 102 of Figure 1 is described in detail on pages 6-14, in accordance with some embodiments. Figure 2 is an exemplary source code representation that may be provided in accordance with claim 1. The source code representation illustrated by Figure 2 is described in greater detail on pages 14-15, in accordance with some embodiments. Figure 5 illustrates an embodiment of the enhanced web services of claim 1. The embodiment is described in greater detail on pages 21-23. Figure 6 illustrates a flowchart describing the operations of claim 1, in accordance with various embodiments. The flowchart is described in greater detail on pages 23-24.

Independent claim 31 is directed towards *an apparatus* which, in substance, is claim 1 in apparatus form. Therefore, support can be found in the same figures and passages in the specification enumerated in the immediately preceding paragraph. Further, additional support can be found in Figure 10 and its corresponding description on page 30. Figure 10 illustrates an exemplary computer system capable of performing the operations recited in claim 1, in accordance with some embodiments.

Independent claims 16 and 23 are directed towards additional methods having similar limitations to those found in claim 1. Therefore, support can be found in the same figures and passages in the specification enumerated in the above paragraphs.

Independent claims 38 and 45 are directed towards additional apparatuses having similar limitations to those found in claims 16 and 23, respectively. Therefore, support can be found in the same figures and passages in the specification enumerated in the above paragraphs.

Grounds For Rejection To Be Argued On Appeal:

- I. Claims 1-52 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 3-22, 24-30, 22, 35-38, 41-63, 66-75, 77-80, 83, and 84 of copending Application No. 10/082,807 (hereinafter '807), which was filed on the same day as the instant application and is owned by the same entity.
- II. Claims 1-4, 10-12, 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46, and 48 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-8, 19-23, 26, 27, 31-36, 38, 39, 43, and 44 of copending Application No. 10/784,492 (hereinafter '492), which was filed after the instant application.
- III. Claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 stand rejected under 35 U.S.C. §102(b) as being anticipated by "Using WebLogic Enterprise JavaBeans" by BEA Systems (hereinafter "BEA WebLogic").
- IV. Claims 2 and 3 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44, and further in view of "EJBDoclet," December 21, 2000, by dreamBean Software (hereinafter "dreamBean").
- V. Claims 5-8, 18, 23-25, 28-30, 40, 45-47, and 50-52 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44, and further in view of "Enterprise JavaBeans" by Monson-Haefel (hereinafter "Monson-Haefel").
- VI. Claims 9, 19, and 41 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44, and further in view of U.S. Patent No. 5,812,768 to *Pagé et al.* (hereinafter "Pagé").
- VII. Claims 12, 31, and 34 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44, and further in view of U.S. Patent No. 6,230,160 to *Chan et al.* (hereinafter "Chan").

VIII. Claims 13, 20, and 42 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44, and further in view of the "Background of the Invention" section appearing on pages 1-3 of the originally filed specification (hereinafter "Background").

IX. Claim 14 stands rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic and Background as applied to claims 13, 20, and 42, and further in view of Pagé.

X. Claims 15, 21, 26, 27, 43, 48, and 49 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic and Background as applied to claims 13, 20, and 42, and further in view of Monson-Haefel.

XI. Claims 32 and 33 stand rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic and Chan as applied to claims 12, 31, and 34, and further in view of dreamBean.

XII. Claim 35 stands rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic and Chan as applied to claims 12, 31, and 34, and further in view of Background.

XIII. Claim 36 stands rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic, Chan, and Background as applied to claim 35, and further in view of Pagé.

XIV. Claim 37 stands rejected under 35 U.S.C. §103(a) over the teachings of BEA WebLogic, Chan, and Background as applied to claim 36, and further in view of Monson-Haefel.

Arguments:

- I. Provisional rejection of claims 1-52 under the judicially created doctrine of obviousness-type double patenting was improper because claims 1-52 are patentably distinct from the claims of '807.

The Examiner has provisionally rejected claims 1-52 as not patentably distinct from the claims of '807 in both the Office Action mailed February 9, 2005 and the final

Office Action mailed August 10, 2005. Thus, the requisite finality for appeal is established.

Applicants respectfully disagree with the Examiner as the claims of '807 are explicitly drawn towards an annotation based development platform for asynchronous web services and not "stateful" web services as indicated in the claims of the instant application. More specifically, exemplary distinctions include that "stateful" web services would include a series of related web service requests, while "asynchronous" web services require coordination as they do not return immediate results.

When a double patenting rejection is appropriate, it must be based either on statutory grounds or nonstatutory grounds. The ground of rejection employed depends upon the relationship of the inventions being claimed. In the instant case, the above-identified Office Action indicates that a provisional nonstatutory obviousness-type double patenting rejection is being asserted. '807 is improperly identified as possessing conflicting claims, which are not identical, but are allegedly not patentably distinct from the claims of the instant application.

Obviousness-type double patenting should only reject application claims when the claimed subject matter is not patentably distinct from the subject matter claimed in a commonly owned patent when the issuance of a second patent would provide unjustified extension of the term of the right to exclude granted by a patent. See *Eli Lilly & Co. v. Barr Labs., Inc.*, 251 F.3d 955, 58 USPQ2d 1865 (Fed. Cir. 2001); *Ex parte Davis*, 56 USPQ2d 1434, 1435-36 (Bd. Pat. App. & Inter. 2000). In the instant case, the claims of the '807 are drawn to "an asynchronous web service" as recited in claim 1 of '807 and not a "stateful web service" as recited in claim 1 of the instant application. Similar language distinguishing the two applications persists throughout. Withdrawal of the provisional rejection is respectfully requested.

Should the Board determine that the distinction between stateful web services and asynchronous web services is "not patentably distinct" as asserted by the Applicants, the Applicants will, upon issuance of either '807 or the instant application, submit the necessary Terminal Disclaimer for the remaining application. Thus, there will be no double patenting.

- II. Provisional rejection of claims 1-4, 10-12 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46, and 48 under the judicially created doctrine of obviousness-type double patenting was improper because claims 1-4, 10-12 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46, and 48 are patentably distinct from the claims of '492.

The Examiner has provisionally rejected claims 1-4, 10-12 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39, 41, 44-46, and 48 as not patentably distinct from the claims of '492 in both the Office Action mailed February 9, 2005 and the final Office Action mailed August 10, 2005. Thus, the requisite finality for appeal is established.

Applicants respectfully disagree with the Examiner's provisional obviousness-type patenting rejection. Although claims of the instant application may dominate the claims of the '492 application and vice versa, this domination in and of itself does not mandate a double patenting rejection. The claims of '492 are specifically directed towards creating network-based software services using source code annotations. In fact, '492 adds Figures 11-15 and claims "an enhanced compiler capable of analyzing the annotated source code, recognizing numerous types of meta-data annotationism and generating a mechanism, which can include one or more of object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component." Although the compiler of '492 may be included in at least one embodiment of the instant application as described by the method claims, this does not mean that the compiler described in '492 is not patentably distinct from the instant

application. As such, the claims of the instant application may dominate '492 without creating a double patenting issue. Thus, Applicants respectfully assert that the claims of the present invention are "patentably distinct" from those of '492.

Domination and double patenting should not be confused. They are two separate issues. One application may "dominate" a second patent or application when the first application has a broad or generic claim which fully encompasses or reads on an invention defined in a narrower or more specific claim in another patent or application. MPEP § 804 (II.) clarifies that domination by itself, i.e., in the absence of statutory or nonstatutory double patenting grounds, cannot support a double patenting rejection. In *re Kaplan*, 789 F.2d 1574, 1577-78, 229 USPQ 678, 681 (Fed. Cir. 1986); and In *re Sarrett*, 327 F.2d 1005, 1014-15, 140 USPQ 474, 482 (CCPA 1964). In the instant case, some of the claims of the instant application may dominate ones of the copending '492 application, but if as in the instant case, the applications claim "patentably distinct" subject matter, such as overlapping distinct methods and systems, there need not necessarily be a double patenting type rejection made. In fact, the very public policy that creates the judicially created nonstatutory double patenting rejection to prevent an inventor from unjustified or improper timewise extension warrants that patentably distinct items not be refused their own patent. Withdrawal of the provisional rejection is respectfully requested.

Should the Board determine that the claims of '492 and those of the instant application are "not patentably distinct" as asserted by the Applicants, the Applicants will, upon issuance of either '492 or the instant application, submit the necessary Terminal Disclaimer for the remaining application. Thus, there will be no double patenting.

III. Rejection of claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 under 35 U.S.C. §102(b) was improper because BEA WebLogic fails to anticipate the claimed invention as claimed in claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44.

It is well settled that anticipation under 35 U.S.C. §102 requires the disclosure in a single piece of prior art to teach **each and every** limitation of a claimed invention. *Electro Med. Sys. S.A. v. Cooper Life Sciences*, 34 F.3d 1048, 1052, 32 USPQ2d 1017, 1019 (Fed. Cir. 1994). . MPEP 2131 states, "TO ANTICIPATE A CLAIM, THE REFERENCE MUST TEACH EVERY ELEMENT OF THE CLAIM" and "a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Furthermore, anticipation requires that each claim element must be identical to a corresponding element in the applied reference. *Glaverbel Société Anonyme v. Northlake Mktg & Supply, Inc.*, 45 F.3d 1550, 1554 (Fed. Cir. 1995). Thus, to anticipate the present invention, BEA WebLogic must disclose every element recited in the pending claims.

Claim 1 calls for a method of specifying a stateful web service within a procedural programming environment, the method comprising:

providing a source code representation of at least a portion of web service logic,

the logic including one or more methods;

identifying one of said one or more methods to be exposed as part of the stateful web service; and

specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified method.

In contrast, BEA WebLogic discloses “encapsulating business logic” in an Enterprise JavaBean, but also requires the developer to “write or obtain an EJBBean” and then advises the developer to “pay careful attention” to conform to the different responsibilities of the “EJB specification”, “the EJBBean writer”, and “EJB framework”. Moreover, BEA WebLogic indicates the need for the developer “to examine packaged EJBBeans and determine if they follow specified relationships” via a utility called compliancechecker. Thus, under BEA WebLogic, ALL of the steps must be performed by the developer. In contrast, the claimed invention, in claim 1, explicitly relies on “a compiler to generate one or more persistent components.” Therefore, the generating of at least some of the persistent components can not be performed by a developer. Moreover, BEA WebLogic clearly indicates that the business logic is encapsulated “inside a component framework” not “exposed as part of the stateful web service” as recited in claim 1 of the instant application.

In further contrast to BEA WebLogic, the instant invention, as claimed in claim 1, only requires “a source code representation of at least a portion of web service logic.” Moreover, a compiler of claim 1 can use one or more declarative annotations “to generate one or more persistent components to maintain conversational state related to the identified method.”

Additionally, in a specific application of Java to the claims at issue, BEA WebLogic does not teach or suggest using an enhanced “compiler to generate” automatically one or more Enterprise JavaBeans™ as well as associated deployment descriptors to store and manage such conversational states based at least in part on “one or more declarative annotations,” as recited in claim 1 of the instant application.

Accordingly, BEA WebLogic does not show, teach or suggest providing “a source code representation of at least a portion of web service logic” and exposing a portion of the logic “as part of the stateful web service” as recited in claim 1 of the present invention.

Claim 16 contains similar language to claim 1, and accordingly is patentable over BEA WebLogic for at least the reasons provided above. And in addition to those reasons, BEA WebLogic does not teach or suggest “parsing...source code to identify the presence of one or more declarative annotations” prior to “generating...one or more object codes...based at least in part upon the source code” and “generating ... meta-data based at least in part upon the one or more declarative annotations” as recited in claim 16.

Claim 38 contains similar language to claims 1 and 16, and is also accordingly patentable over BEA WebLogic for at least the reasons provided above.

Claims 4, 10, 11, 17, 22, 39, and 44 depend from claims 1, 16, and 38, incorporating their limitations respectively. Accordingly, for at least the same reasons, claims 4, 10, 11, 17, 22, 39, and 44 are patentable over BEA WebLogic.

- IV. Rejection of claims 2 and 3 under 35 U.S.C. §103(a) was improper because BEA WebLogic and dreamBean, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 2 and 3 is viewed as a whole.

dreamBean does not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claim 1 remains patentable over BEA WebLogic even when

combined with dreamBean.

Claims 2 and 3 depend on claim 1, incorporating its limitations respectively. Therefore, for at least the same reasons, Claims 2 and 3 are patentable over BEA WebLogic and dreamBean, alone or in combination.

- V. Rejection of claims 5-8, 18, 23-25, 28-30, 40, 45-47, and 50-52 under 35 U.S.C. §103(a) was improper because BEA WebLogic and Monson-Haefel, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 5-8, 18, 23-25, 28-30, 40, 45-47, and 50-52 is viewed as a whole.

Monson-Haefel does not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claims 1, 16, 23, 38, and 45 (claims 23 and 45 are independent claims containing similar language to claims 1, 16, and 38) remain patentable over BEA WebLogic even when combined with Monson-Haefel.

Claims 5-8, 18, 24-25, 28-30, 40, 46-47, and 50-52 depend on claims 1, 16, 23, 38, and 45, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 5-8, 18, 24-25, 28-30, 40, 46-47, and 50-52 are patentable over BEA WebLogic and Monson-Haefel, alone or in combination.

- VI. Rejection of claims 9, 19, and 41 under 35 U.S.C. §103(a) was improper because BEA WebLogic and Pagé, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 9, 19, and 41 is viewed as a whole.

Pagé does not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claims 1, 16, and 38 remain patentable over BEA WebLogic and Pagé, alone

or in combination.

Claims 9, 19, and 41 depend on claims 1, 16, and 38, incorporating their limitations respectively. Therefore, for at least the same reasons, claims 26-31 are patentable over BEA WebLogic and Pagé, alone or in combination.

VII. Rejection of claims 12, 31, and 34, under 35 U.S.C. §103(a) was improper because BEA WebLogic and Chan, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 12, 31, and 34 is viewed as a whole.

Chan does not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claims 1 and 31 (claim 31 is an independent claim containing similar language to claims 1, 16, and 38) remain patentable over BEA WebLogic and Chan, alone or in combination.

Claims 12 and 34 depend on either Claims 1 or 31, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 12 and 34 are patentable over BEA WebLogic and Chan, alone or in combination.

VIII. Rejection of claims 13, 20, and 42 under 35 U.S.C. §103(a) was improper because BEA WebLogic and Background, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 13, 20, and 42 is viewed as a whole.

Background does not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claims 1, 16, and 38 remain patentable over BEA WebLogic and Background, alone or in combination.

Claims 13, 20, and 42 depend on claims 1, 16, and 38, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 13, 20, and 42 are patentable over BEA WebLogic and Background, alone or in combination.

- IX. Rejection of claim 14 under 35 U.S.C. §103(a) was improper because BEA WebLogic, Background, and Pagé, alone or in combination, fail to teach the claimed invention when the invention as claimed in claim 14 is viewed as a whole.

Background and Pagé, alone or in combination, do not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claim 1 remains patentable over BEA WebLogic, Background, and Pagé, alone or in combination.

Claim 14 depends on claim 1 incorporating its limitations respectively. Therefore, for at least the same reasons, Claim 14 is patentable over BEA WebLogic, Background, and Pagé, alone or in combination.

- X. Rejection of claims 15, 21, 26, 27, 43, 48, and 49 under 35 U.S.C. §103(a) was improper because BEA WebLogic, Background, and Monson-Haefel, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 15, 21, 26, 27, 43, 48, and 49 is viewed as a whole.

Background and Monson-Haefel, alone or in combination, do not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claims 1, 16, 23, 38, and 45 remain patentable over BEA WebLogic, Background, and Monson-Haefel, alone or in combination.

Claims 15, 21, 26, 27, 43, 48, and 49 depend on claims 1, 16, 23, 38, and 45, incorporating their limitations respectively. Therefore, for at least the same reasons, Claims 15, 21, 26, 27, 43, 48, and 49 are patentable over BEA WebLogic, Background, and Monson-Haefel, alone or in combination.

XI. Rejection of claims 32 and 33 under 35 U.S.C. §103(a) was improper because BEA WebLogic, Chan, and dreamBean, alone or in combination, fail to teach the claimed invention when the invention as claimed in claims 32 and 33 is viewed as a whole.

Chan and dreamBean, alone or in combination, do not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claim 31 remains patentable over BEA WebLogic, Chan, and dreamBean, alone or in combination.

Claims 32 and 33 depend on claim 31, incorporating its limitations respectively. Therefore, for at least the same reasons, Claims 32 and 33 are patentable over BEA WebLogic, Chan, and dreamBean, alone or in combination.

XII. Rejection of claim 35 under 35 U.S.C. §103(a) was improper because BEA WebLogic, Chan, and Background, alone or in combination, fail to teach the claimed invention when the invention as claimed in claim 35 is viewed as a whole.

Chan and Background, alone or in combination, do not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claim 31 remains patentable over BEA WebLogic, Chan, and Background, alone or in combination.

Claim 35 depends on claim 31, incorporating its limitations respectively. Therefore, for at least the same reasons, Claim 35 is patentable over BEA WebLogic, Chan, and Background, alone or in combination.

XIII. Rejection of claim 36, under 35 U.S.C. §103(a) was improper because BEA WebLogic, Chan, Background, and Pagé, alone or in combination, fail to teach the claimed invention when the invention as claimed in claim 36 is viewed as a whole.

Chan, Background, and Pagé, alone or in combination, do not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claim 31 remains patentable over BEA WebLogic, Chan, Background, and Pagé, alone or in combination.

Claim 36 depends on claim 31, incorporating its limitations respectively. Therefore, for at least the same reasons, Claim 36 is patentable over BEA WebLogic, Chan, Background, and Pagé, alone or in combination.

XIV. Rejection of claim 37 under 35 U.S.C. §103(a) was improper because BEA WebLogic, Chan, Background, and Monson-Haefel, alone or in combination, fail to teach the claimed invention when the invention as claimed in claim 37 is viewed as a whole.

Chan, Background, and Monson-Haefel, alone or in combination, do not remedy the above-discussed deficiencies of BEA WebLogic. Therefore, claim 31 remains patentable over BEA WebLogic, Chan, Background, and Monson-Haefel, alone or in combination.

Claim 37 depends on claim 31, incorporating its limitations respectively. Therefore, for at least the same reasons, Claim 37 is patentable over BEA WebLogic, Chan, Background, and Monson-Haefel, alone or in combination.

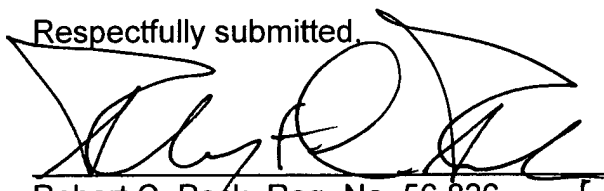
Conclusion

Appellant respectfully submits that all the appealed claims in this application are patentable and requests that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims.

This brief is re-submitted in triplicate, along with Check Number 13531 for \$500.00 to cover the filing of appeal brief. We do not believe any additional fees, in particular extension of time fees, are needed. However, should that be necessary, please charge our deposit account 500393. In addition, please charge any shortages and credit any overages to Deposit Account No. 500393.

Date: March 9, 2006

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Robert C. Peek', is written over a horizontal line.

Robert C. Peek, Reg. No. 56,826
Agent for Appellant Applicants

Schwabe Williamson & Wyatt, P.C.
1420 Fifth, Suite 3010
Seattle, WA 98101
Tel: (206) 622-1711
Fax: (206) 292-0460

Appendix A – Appealed Claims

1. (Original) A method of specifying a stateful web service within a procedural programming environment, the method comprising:
 - providing a source code representation of at least a portion of web service logic,
 - the logic including one or more methods;
 - identifying one of said one or more methods to be exposed as part of the stateful web service; and
 - specifying one or more declarative annotations to cause a compiler to generate one or more persistent components to maintain conversational state related to the identified method.
2. (Original) The method of claim 1, wherein the one or more declarative annotations are specified within the source code representation.
3. (Original) The method of claim 2, wherein the one or more declarative annotations are specified within a comment field preceding the identified method.
4. (Original) The method of claim 1, wherein the one or more declarative annotations are specified outside of the source code representation and associated with the identified method by the compiler.
5. (Original) The method of claim 1, wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a

start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between a client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between a client and the web service.

6. (Original) The method of claim 5, wherein when a method declared to be a start method is invoked at run-time, a new instance of a conversation is created, and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

7. (Original) The method of claim 5, wherein when a method declared to be a continue method or a finish method is invoked at run-time, a unique identifier provided by the client is obtained and used to access a corresponding instance of a conversation.

8. (Original) The method of claim 7, wherein when a finish method is invoked at run-time, the corresponding instance of the conversation is destroyed after processing by the web service logic.

9. (Original) The method of claim 1, wherein the one or more declarative annotations indicate to the compiler whether the identified method is buffered, wherein

if the identified method is buffered the compiler instantiates one or more queues to temporarily store one or more requests for the identified method.

10. (Original) The method of claim 1, wherein the one or more declarative annotations are manually specified by a developer.

11. (Original) The method of claim 1, wherein the one or more declarative annotations are automatically specified by an integrated development environment based upon input provided by a developer.

12. (Original) The method of claim 11, wherein said input includes graphical manipulation of the identified method by the developer via the integrated development environment.

13. (Original) The method of claim 1, wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client.

14. (Original) The method of claim 13, wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service.

15. (Original) The method of claim 13, wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.

16. (Previously Presented) In a procedural programming environment, a method of generating a stateful web service, the method comprising:

reading on one or more computing devices a segment of procedural source

code representing at least a portion of the web service;

parsing on one or more computing devices the segment of source code to

identify the presence of one or more declarative annotations identifying

an associated method within the segment as being stateful;

generating on one or more computing devices one or more object codes

defining one or more publicly accessible service components based at

least in part upon the source code;

generating on one or more computing devices meta-data based at least in part

upon the one or more declarative annotations;

associating on one or more computing devices the meta-data with the one or

more object codes.

17. (Previously Presented) The method of claim 16, further comprising generating on one or more computing devices one or more persistent components to maintain conversational state relating the associated method.

18. (Original) The method of claim 16, wherein the one or more declarative annotations further identify the associated method as being at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between a client and the web service, the continue method applies to the continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between a client and the web service.

19. (Original) The method of claim 16, wherein the one or more declarative annotations further identify the associated method as being a buffered method, wherein one or more queues are instantiated to temporarily store one or more requests for the identified method.

20. (Previously Presented) The method of claim 16, further comprising:
generating on one or more computing devices a proxy object designed to facilitate interaction by the web service with an external web service.

21. (Previously Presented) The method of claim 20, further comprising:
generating on one or more computing devices a unique identifier to identify a specific instance of the external web service.

22. (Original) The method of claim 16, wherein the source code is written in the Java programming language.

23. (Original) In a stateful web service, a method comprising:
- receiving a message requesting that a web service method be invoked;
 - parsing the message to identify the requested method;
 - determining whether the method is a stateful method based at least in part upon meta-data derived from one or more declarative annotations stored in association with object codes of the web service; and
 - dispatching the received request to invoke the identified stateful method.
24. (Original) The method of claim 23, wherein the message is received from a remote client.
25. (Original) The method of claim 23, wherein the message includes a globally unique conversational identifier identifying a specific conversational instance to facilitate management of multiple simultaneous conversations by the web service.
26. (Original) The method of claim 23, wherein the message is a SOAP based message.
27. (Original) The method of claim 26, wherein the conversational identifier is a GUID encapsulated in a header of the SOAP message.

28. (Original) The method of claim 23, wherein if a start method is invoked, a new instance of a conversation is created and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

29. (Original) The method of claim 23, wherein if a continue method or a finish method is invoked, a unique identifier provided by the client is identified and used to access a corresponding instance of a conversation.

30. (Original) The method of claim 29, wherein if a finish method is invoked, the corresponding instance of the conversation is destroyed by the web service logic after processing.

31. (Original) An article of manufacture comprising:
a storage medium having stored therein a plurality of programming instructions,
which when executed provide a graphical interface to facilitate
specification of one or more declarative annotations within a procedural
programming environment to modify an identified method of a stateful
web service to cause a compiler to generate one or more persistent
components to maintain conversational state related to the identified
method.

32. (Original) The article of claim 31, wherein the one or more declarative annotations are specified within a source code representation of at least a portion of the web service based at least in part upon graphical input by a developer.

33. (Original) The article of claim 32, wherein the one or more declarative annotations are specified within a comment field preceding the identified method.

34. (Original) The article of claim 31, wherein the one or more declarative annotations are specified outside of a source code representation of at least a portion of the web service and associated with the identified method by the compiler.

35. (Original) The article of claim 31, wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client.

36. (Original) The article of claim 35, wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service.

37. (Original) The article of claim 35, wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.

38. (Original) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to generate a stateful web service, which programming instructions when executed enable the apparatus to:

read a segment of procedural source code representing at least a portion of the web service;

parse the segment of source code to identify the presence of one or more declarative annotations identifying an associated method within the segment as being stateful;

generate one or more object codes defining one or more publicly accessible service components based at least in part upon the source code;

generate meta-data based at least in part upon the one or more declarative annotations;

associate the meta-data with the one or more object codes.

39. (Original) The article of claim 38, wherein the instructions when executed further enable the apparatus to generate one or more persistent components to maintain conversational state relating the associated method.

40. (Original) The article of claim 38, wherein the one or more declarative annotations further identify the associated method as being at least one of a start method, a continue method, and a finish method, wherein the start method applies to the start of a stateful conversation between a client and the web service, the continue

method applies to the continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to the completion of an ongoing stateful conversation between a client and the web service.

41. (Original) The article of claim 38, wherein the one or more declarative annotations further identify the associated method as being a buffered method, wherein one or more queues are instantiated to temporarily store one or more requests for the identified method.

42. (Original) The article of claim 38, wherein the instructions when executed further enable the apparatus to generate a proxy object designed to facilitate interaction by the web service with an external web service.

43. (Original) The article of claim 42, wherein the instructions when executed further enable the apparatus to generate a unique identifier to identify a specific instance of the external web service.

44. (Original) The article of claim 38, wherein the source code is written in the Java programming language.

45. (Original) An article of manufacture comprising:

a storage medium having stored therein a plurality of programming instructions designed to program an apparatus to generate a stateful web service, which programming instructions when executed enable the apparatus to

receive a message requesting that a method of the web service be invoked;

parse the message to identify the requested method;

determine whether the method is a stateful method based at least in part upon meta-data derived from one or more declarative annotations stored in association with object codes of the web service; and

dispatch the received request to invoke the identified stateful method.

46. (Original) The article of claim 45, wherein the message is received from a remote client.

47. (Original) The article of claim 45, wherein the message includes a globally unique conversational identifier identifying a specific conversational instance to facilitate management of multiple simultaneous conversations by the web service.

48. (Original) The article of claim 45, wherein the message is a SOAP based message.

49. (Original) The article of claim 48, wherein the conversational identifier is a GUID encapsulated in a header of the SOAP message.

50. (Original) The article of claim 45, wherein if a start method is invoked, a new instance of a conversation is created and a unique identifier is associated with that conversational instance to facilitate management of multiple simultaneous conversations.

51. (Original) The article of claim 45, wherein if a continue method or a finish method is invoked, a unique identifier provided by the client is identified and used to access a corresponding instance of a conversation.

52. (Original) The article of claim 51, wherein if a finish method is invoked, the corresponding instance of the conversation is destroyed by the web service logic after processing.

Appendix B – Copies of Evidence Submitted

No evidence has been submitted under 37 C.F.R. 1.130, 1.131, or 1.132. No evidence entered by Examiner has been relied upon by Appellants in the appeal.

Appendix C – Related Proceedings

To the best of Appellants' knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.